(12) UK Patent Application (19) GB (11) 2 325 324 (13) A

(43) Date of A Publication 18.11.1998

(21) Application No 9811847.4

(22) Date of Filing 21.02.1997

Date Lodged 02.06.1998

(30) Priority Data
(31) 08619073 (32) 20.03.1996 (33) US

(62) Divided from Application No 9703665.1 under Section 15(4) of the Patents Act 1977

(51) INT CL[6]
G06F 17/50

(52) UK CL (Edition P)
G4H HU H13D H14A
G4A AUB
U1S S2092

(56) Documents Cited
None

(58) Field of Search
UK CL (Edition P) G4A AUB , G4H HU
INT CL[6] G06F

(71) Applicant(s)
Altera Corporation
(Incorporated in USA - California)
2610 Orchard Parkway, Sunnyvale, San Jose, California 95134-2020, United States of America

(72) Inventor(s)
Jan Young Lee

(74) Agent and/or Address for Service
Withers & Rogers
4 Dyer's Buildings, Holborn, LONDON, EC1N 2QP, United Kingdom

(54) Abstract Title
Implementing circuit designs in physical circuits

(57) In order to reduce the amount of work that must be done to implement a revised circuit design in physical circuitry, portions of the new design that are the same as a previously implemented design are identified. This preferably involves several different, successively applied approaches to attempting to find matching circuit components in the new and old designs (names, basic feature sets, and considering immediate neighbours). Connections that have changed between otherwise unchanged components are also identified.

GB 2 325 324 A

FIG. 1a

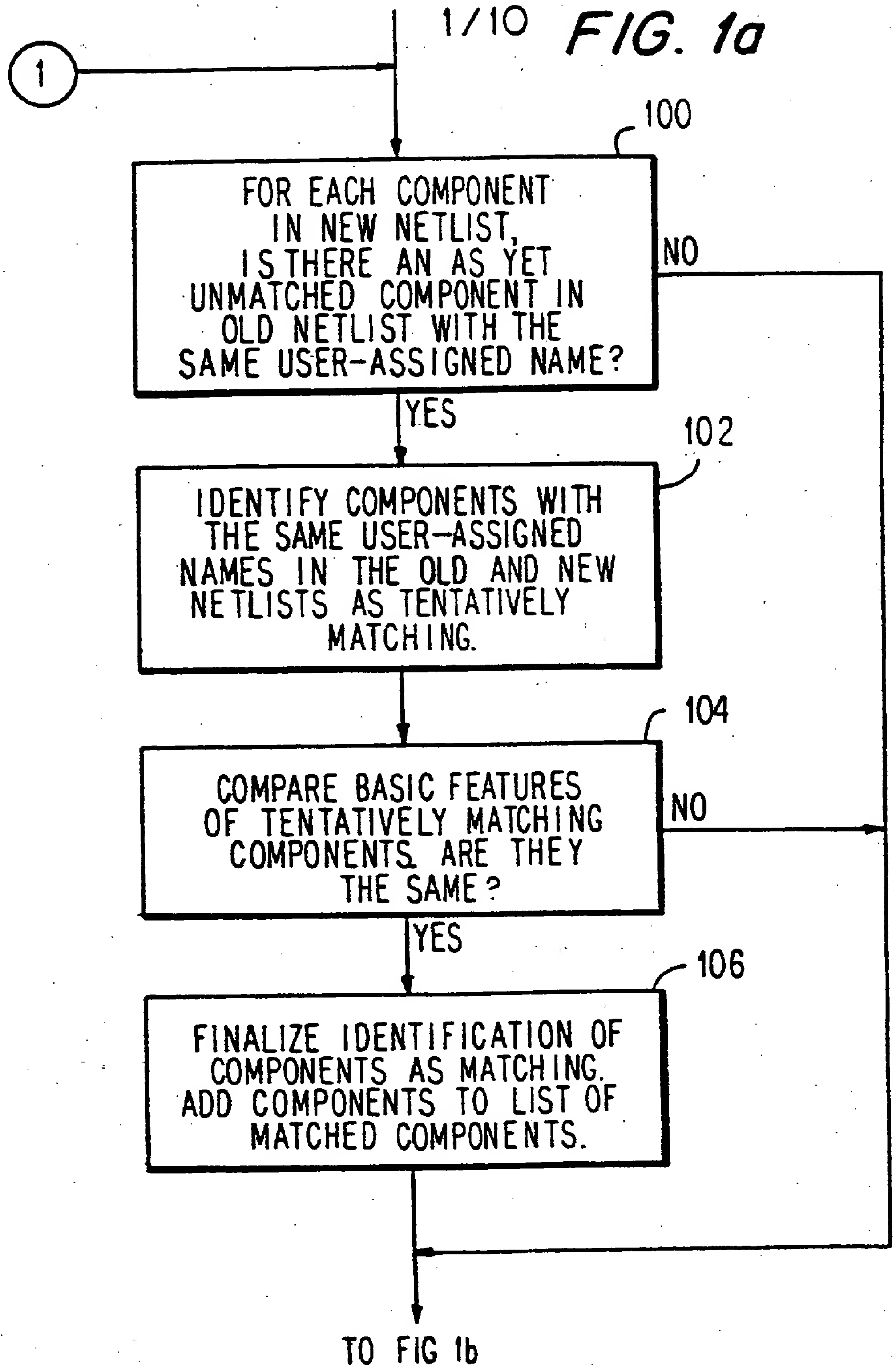

1
100
FOR EACH COMPONENT IN NEW NETLIST, IS THERE AN AS YET UNMATCHED COMPONENT IN OLD NETLIST WITH THE SAME USER-ASSIGNED NAME?
NO
YES
102
IDENTIFY COMPONENTS WITH THE SAME USER-ASSIGNED NAMES IN THE OLD AND NEW NETLISTS AS TENTATIVELY MATCHING.
104
COMPARE BASIC FEATURES OF TENTATIVELY MATCHING COMPONENTS. ARE THEY THE SAME?
NO
YES
106
FINALIZE IDENTIFICATION OF COMPONENTS AS MATCHING. ADD COMPONENTS TO LIST OF MATCHED COMPONENTS.
TO FIG 1b

FIG. 1b

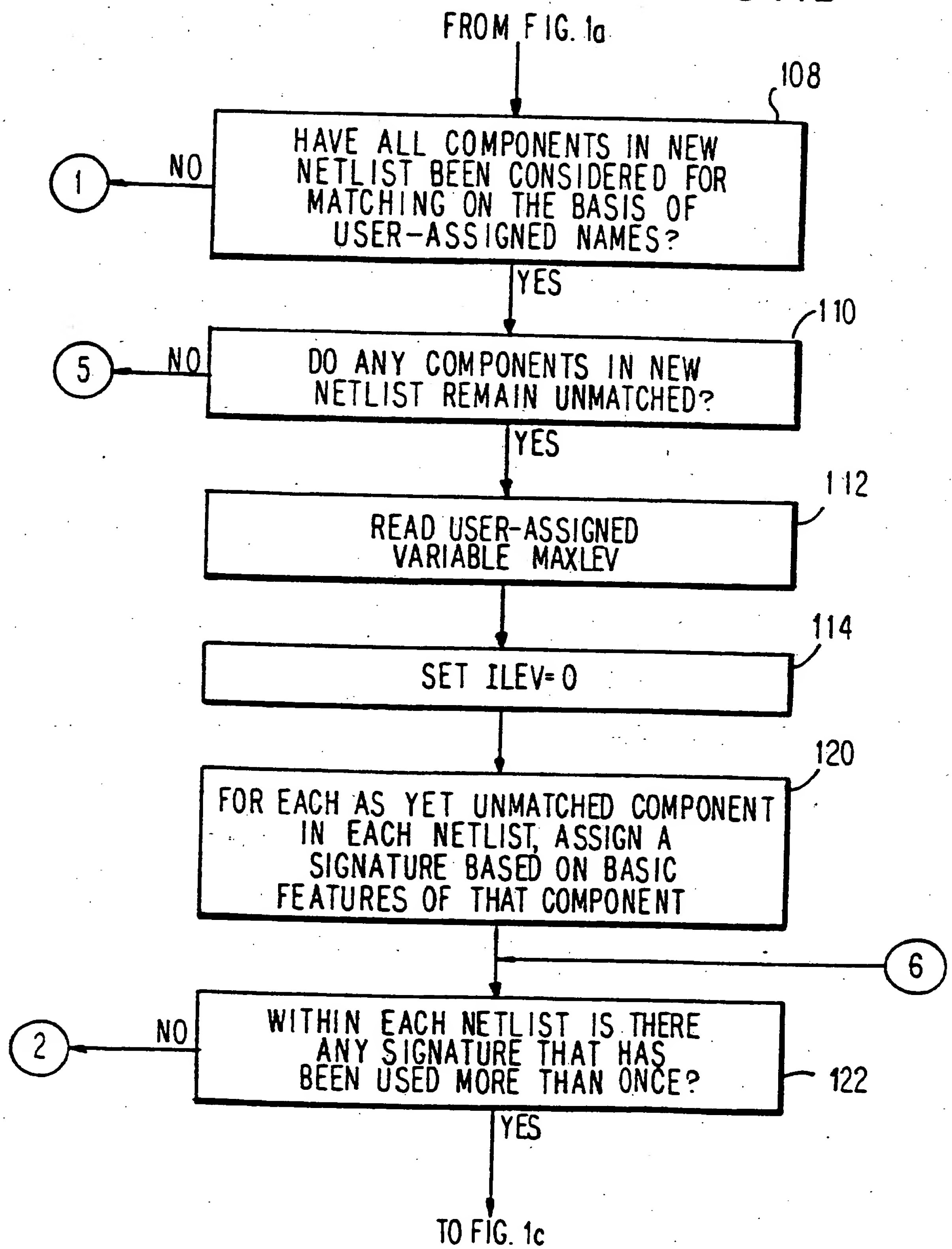

FROM FIG. 1a
108
HAVE ALL COMPONENTS IN NEW NETLIST BEEN CONSIDERED FOR MATCHING ON THE BASIS OF USER-ASSIGNED NAMES?
NO
1
YES
110
DO ANY COMPONENTS IN NEW NETLIST REMAIN UNMATCHED?
NO
5
YES
112
READ USER-ASSIGNED VARIABLE MAXLEV
114
SET ILEV=0
120
FOR EACH AS YET UNMATCHED COMPONENT IN EACH NETLIST, ASSIGN A SIGNATURE BASED ON BASIC FEATURES OF THAT COMPONENT
6
WITHIN EACH NETLIST IS THERE ANY SIGNATURE THAT HAS BEEN USED MORE THAN ONCE?
NO
2
122
YES
TO FIG. 1c

FIG. 1c

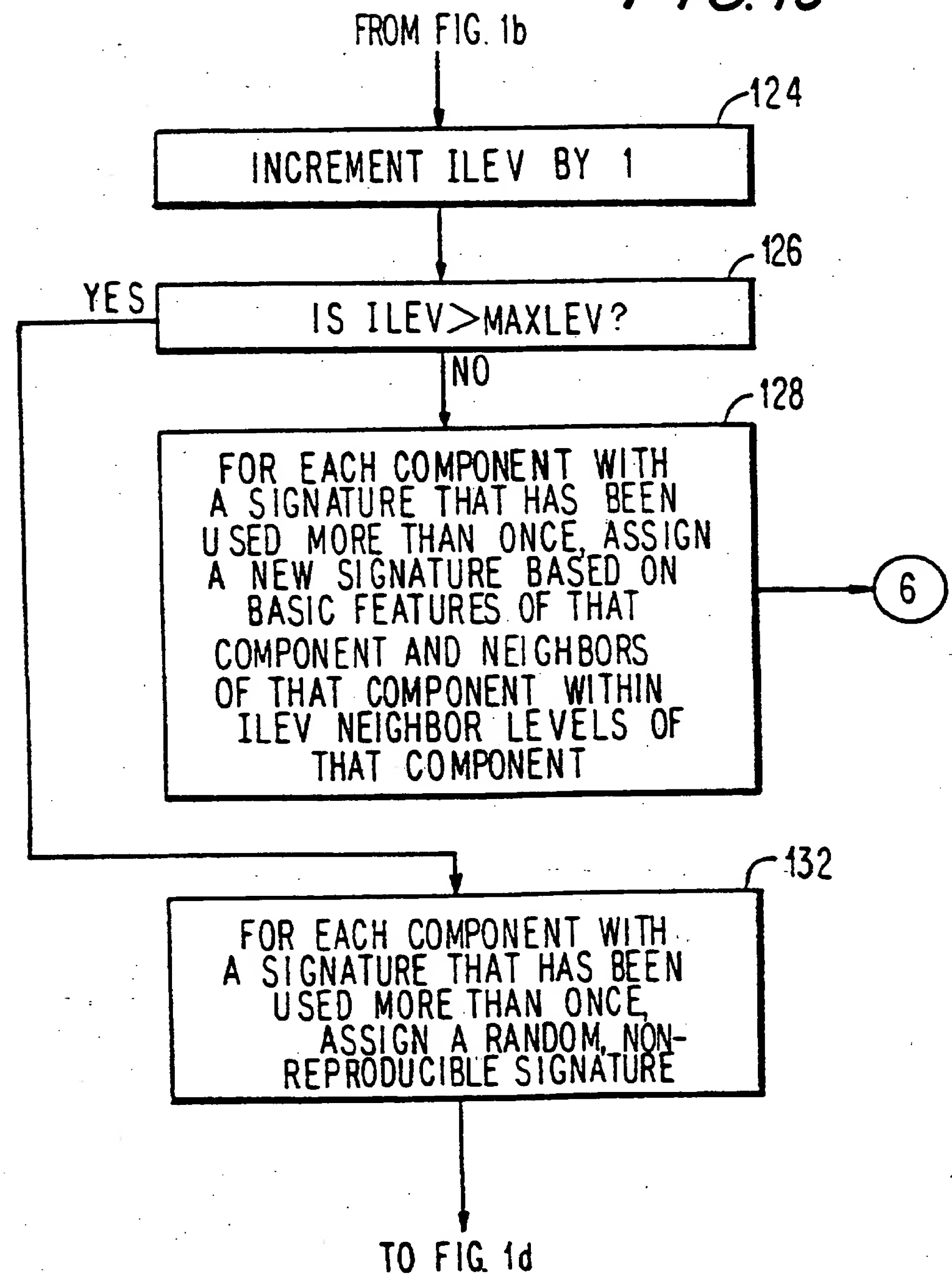

FROM FIG. 1b
124
INCREMENT ILEV BY 1
126
YES
IS ILEV>MAXLEV?
NO
128
FOR EACH COMPONENT WITH A SIGNATURE THAT HAS BEEN USED MORE THAN ONCE, ASSIGN A NEW SIGNATURE BASED ON BASIC FEATURES OF THAT COMPONENT AND NEIGHBORS OF THAT COMPONENT WITHIN ILEV NEIGHBOR LEVELS OF THAT COMPONENT
6
132
FOR EACH COMPONENT WITH A SIGNATURE THAT HAS BEEN USED MORE THAN ONCE, ASSIGN A RANDOM, NON-REPRODUCIBLE SIGNATURE
TO FIG. 1d

FIG. 1d

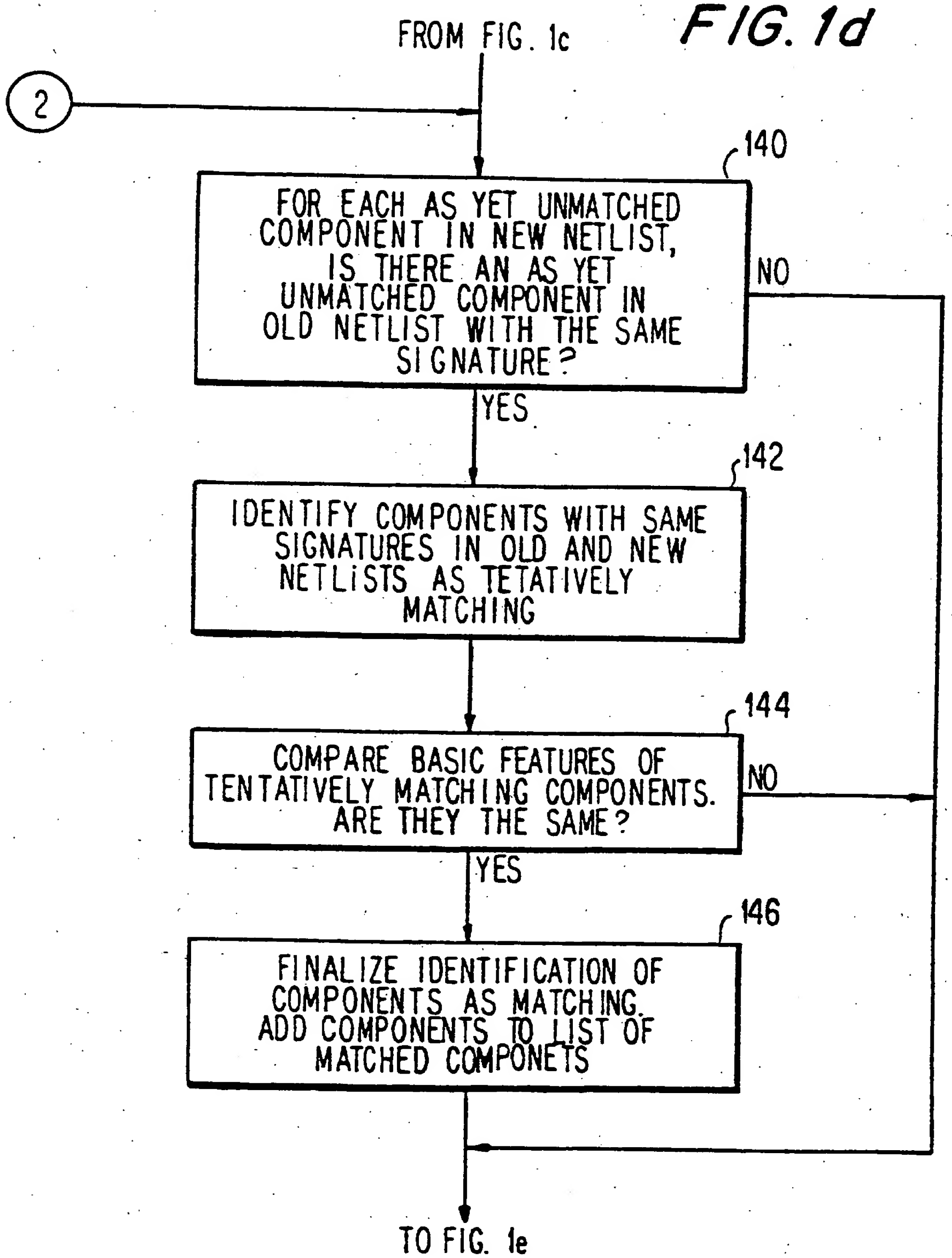

FROM FIG. 1c
2
140
FOR EACH AS YET UNMATCHED COMPONENT IN NEW NETLIST, IS THERE AN AS YET UNMATCHED COMPONENT IN OLD NETLIST WITH THE SAME SIGNATURE?
NO
YES
142
IDENTIFY COMPONENTS WITH SAME SIGNATURES IN OLD AND NEW NETLISTS AS TETATIVELY MATCHING
144
COMPARE BASIC FEATURES OF TENTATIVELY MATCHING COMPONENTS. ARE THEY THE SAME?
NO
YES
146
FINALIZE IDENTIFICATION OF COMPONENTS AS MATCHING. ADD COMPONENTS TO LIST OF MATCHED COMPONETS
TO FIG. 1e

FIG. 1e

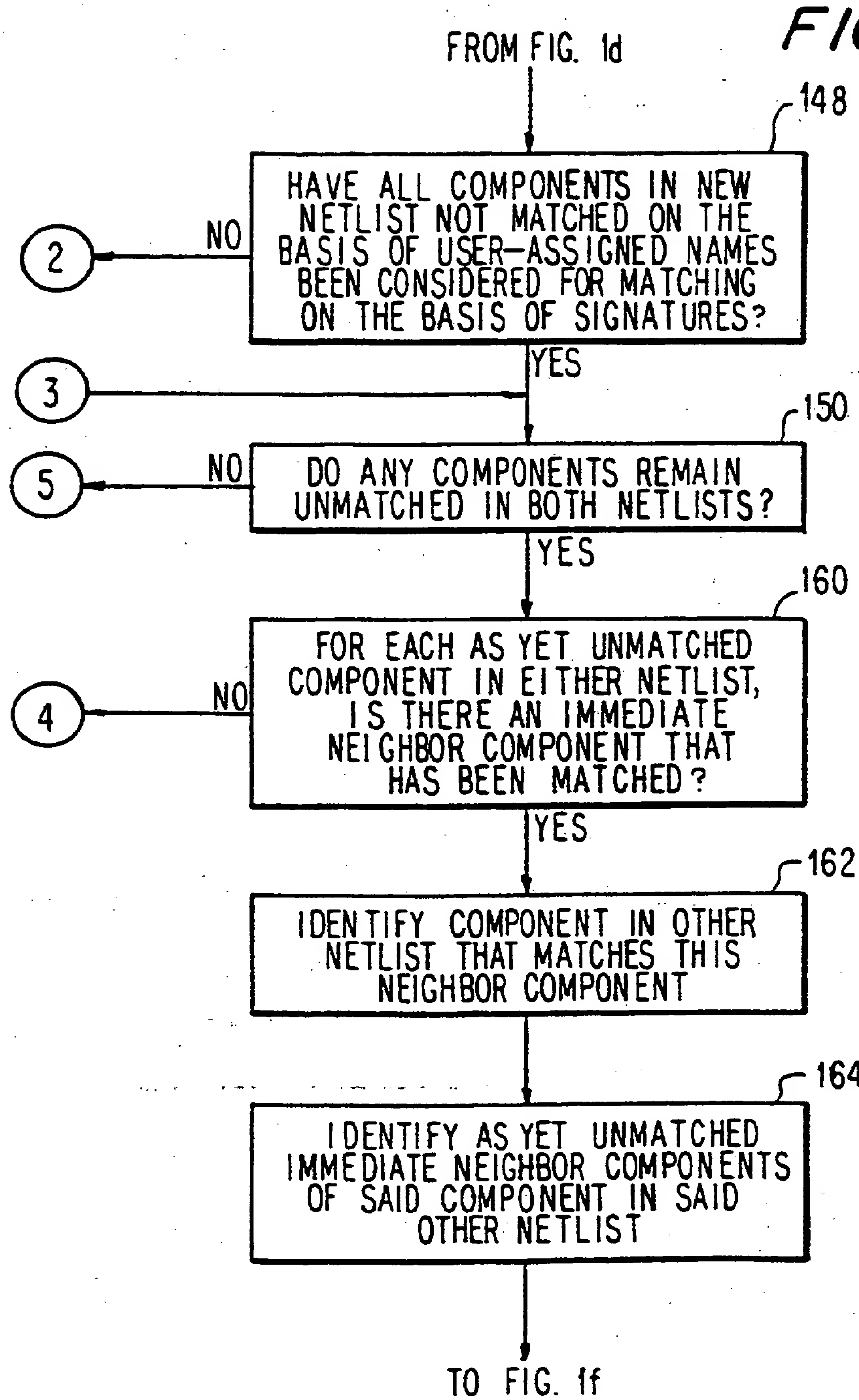

FROM FIG. 1d
148
HAVE ALL COMPONENTS IN NEW NETLIST NOT MATCHED ON THE BASIS OF USER-ASSIGNED NAMES BEEN CONSIDERED FOR MATCHING ON THE BASIS OF SIGNATURES?
NO
2
YES
3
150
DO ANY COMPONENTS REMAIN UNMATCHED IN BOTH NETLISTS?
NO
5
YES
160
FOR EACH AS YET UNMATCHED COMPONENT IN EITHER NETLIST, IS THERE AN IMMEDIATE NEIGHBOR COMPONENT THAT HAS BEEN MATCHED?
NO
4
YES
162
IDENTIFY COMPONENT IN OTHER NETLIST THAT MATCHES THIS NEIGHBOR COMPONENT
164
IDENTIFY AS YET UNMATCHED IMMEDIATE NEIGHBOR COMPONENTS OF SAID COMPONENT IN SAID OTHER NETLIST
TO FIG. 1f

FIG. 1f

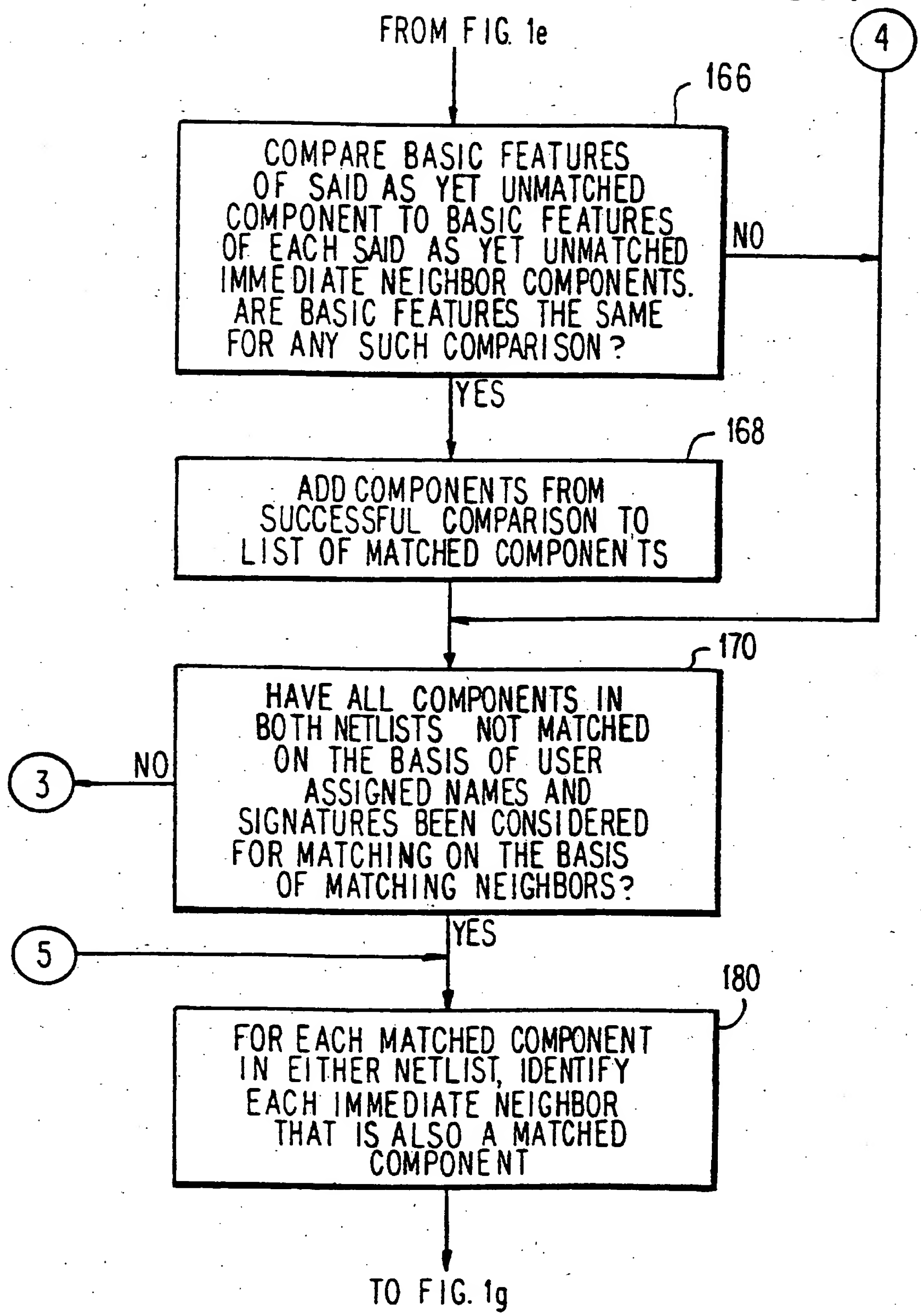
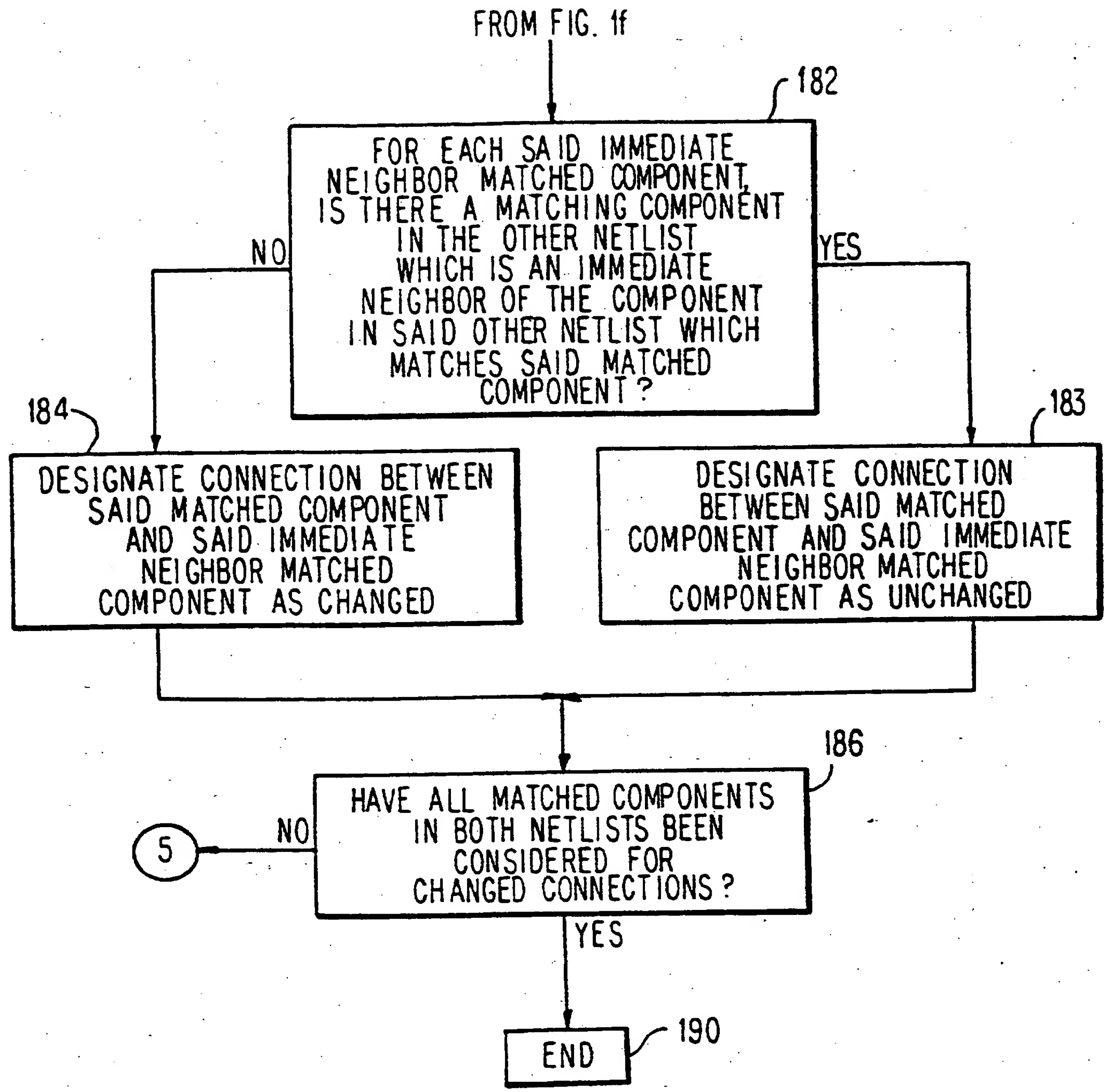

FROM FIG. 1e
4
166
COMPARE BASIC FEATURES OF SAID AS YET UNMATCHED COMPONENT TO BASIC FEATURES OF EACH SAID AS YET UNMATCHED IMMEDIATE NEIGHBOR COMPONENTS. ARE BASIC FEATURES THE SAME FOR ANY SUCH COMPARISON?
NO
YES
168
ADD COMPONENTS FROM SUCCESSFUL COMPARISON TO LIST OF MATCHED COMPONENTS
170
HAVE ALL COMPONENTS IN BOTH NETLISTS NOT MATCHED ON THE BASIS OF USER ASSIGNED NAMES AND SIGNATURES BEEN CONSIDERED FOR MATCHING ON THE BASIS OF MATCHING NEIGHBORS?
NO
3
YES
5
180
FOR EACH MATCHED COMPONENT IN EITHER NETLIST, IDENTIFY EACH IMMEDIATE NEIGHBOR THAT IS ALSO A MATCHED COMPONENT
TO FIG. 1g

FIG. 1g


FROM FIG. 1f
182
FOR EACH SAID IMMEDIATE NEIGHBOR MATCHED COMPONENT, IS THERE A MATCHING COMPONENT IN THE OTHER NETLIST WHICH IS AN IMMEDIATE NEIGHBOR OF THE COMPONENT IN SAID OTHER NETLIST WHICH MATCHES SAID MATCHED COMPONENT?
NO
YES
184
DESIGNATE CONNECTION BETWEEN SAID MATCHED COMPONENT AND SAID IMMEDIATE NEIGHBOR MATCHED COMPONENT AS CHANGED
183
DESIGNATE CONNECTION BETWEEN SAID MATCHED COMPONENT AND SAID IMMEDIATE NEIGHBOR MATCHED COMPONENT AS UNCHANGED
186
HAVE ALL MATCHED COMPONENTS IN BOTH NETLISTS BEEN CONSIDERED FOR CHANGED CONNECTIONS?
5
NO
YES
END
190

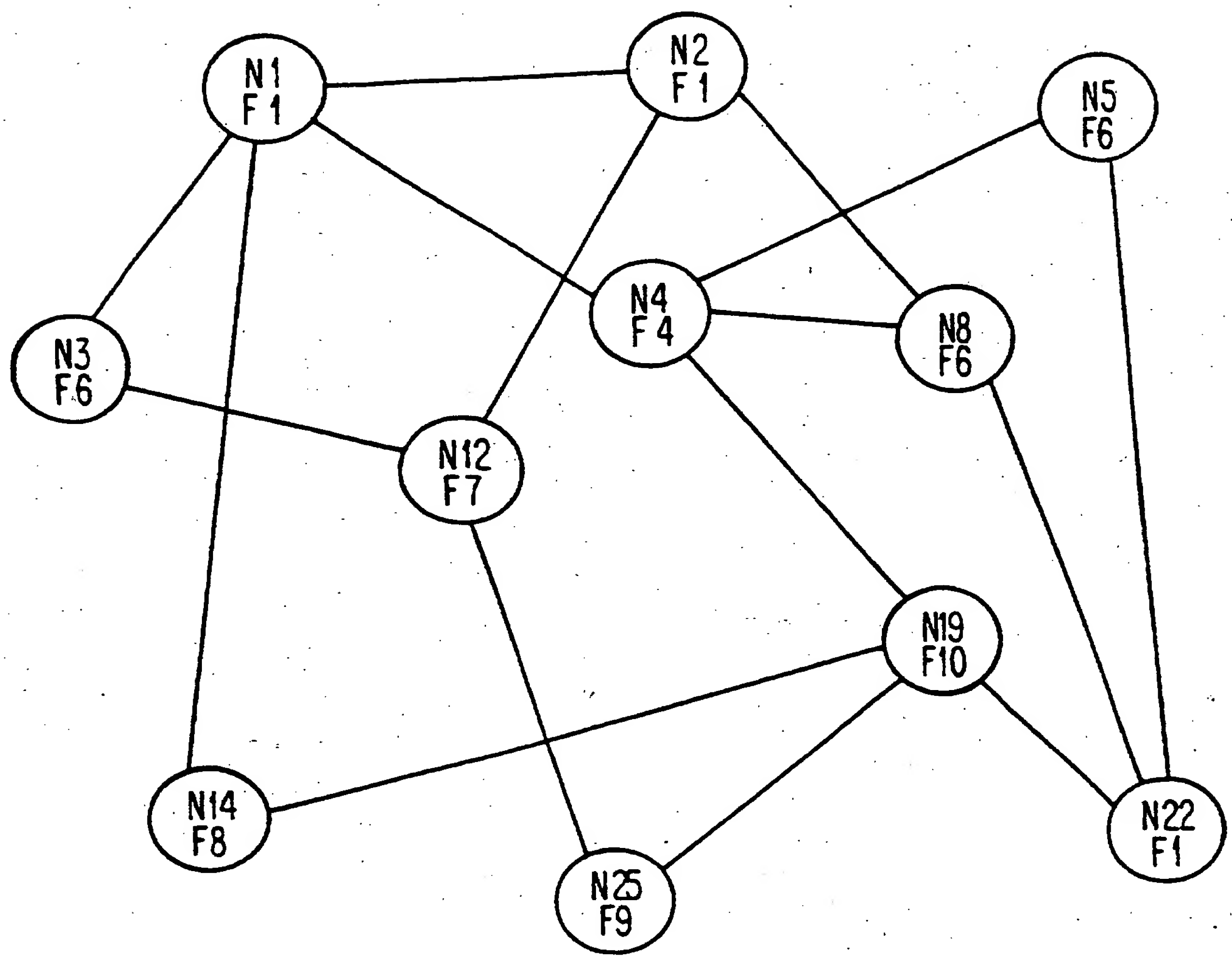

N1
F1
N2
F1
N5
F6
N3
F6
N4
F4
N8
F6
N12
F7
N19
F10
N14
F8
N25
F9
N22
F1

FIG. 2a

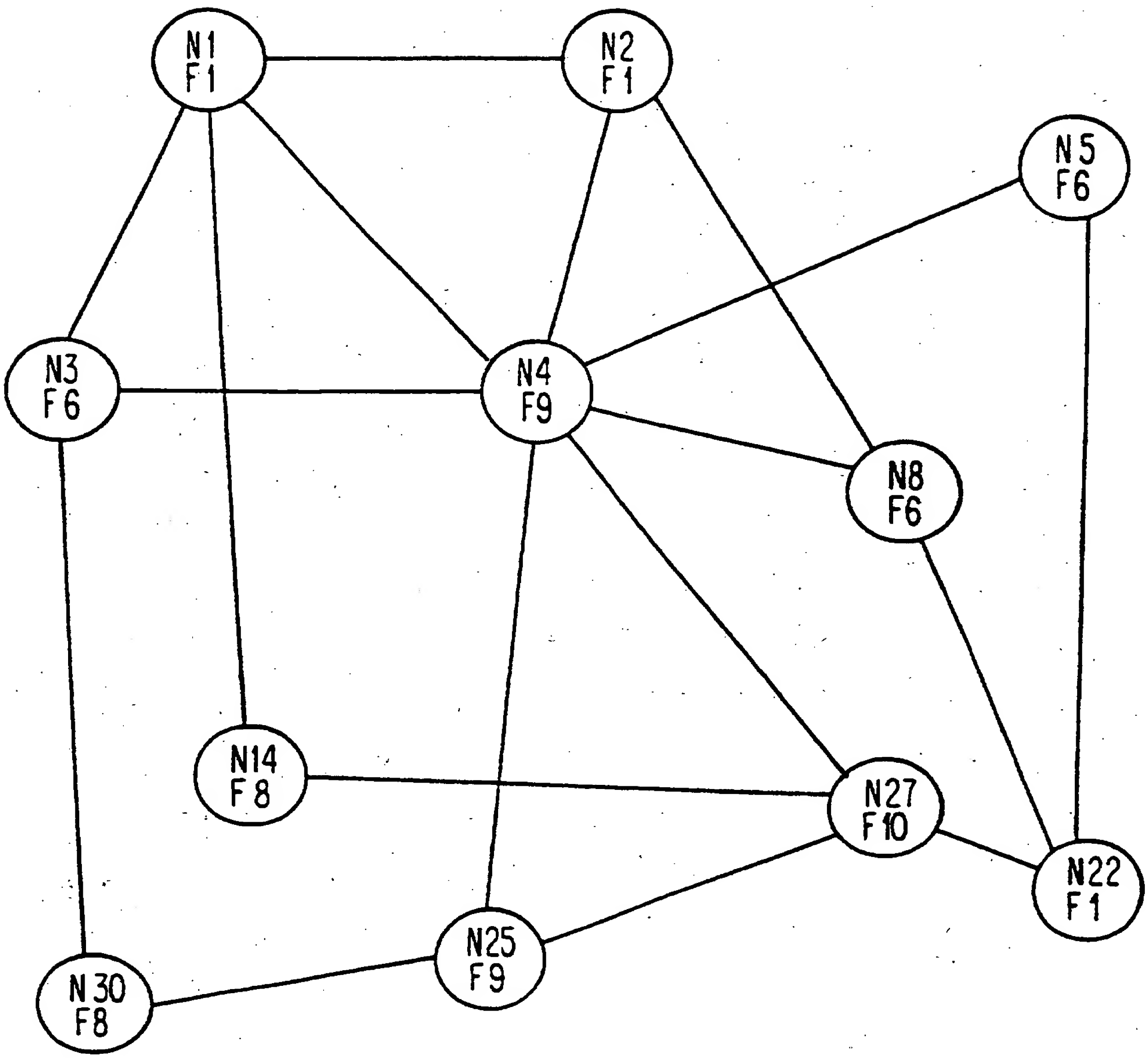

N1
F1
N2
F1
N5
F6
N3
F6
N4
F9
N8
F6
N14
F8
N27
F10
N22
F1
N25
F9
N30
F8

FIG. 2b

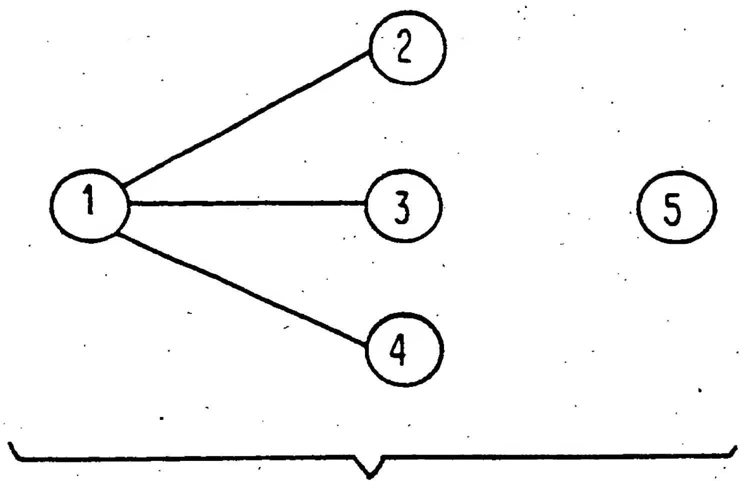


FIG.3a

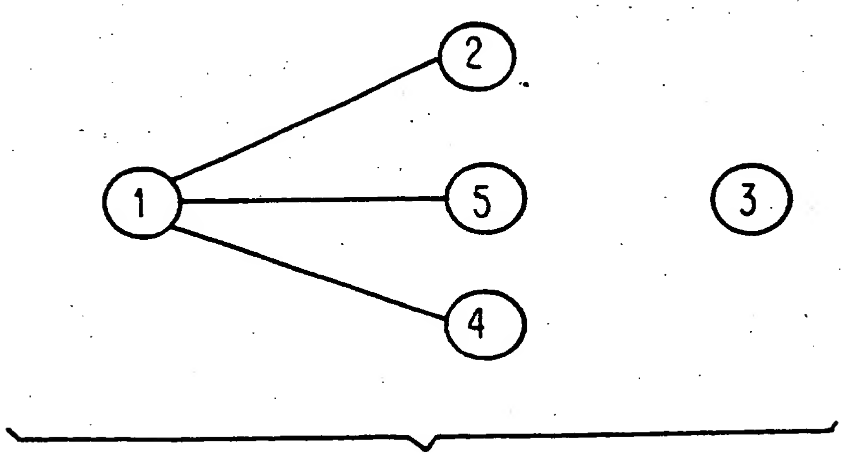


FIG.3b

2325324

METHODS FOR IMPLEMENTING CIRCUIT DESIGNS IN PHYSICAL CIRCUITS

Background of the Invention

This invention relates to methods for implementing circuit designs in physical circuits, and more particularly to comparing two circuit designs so that as much as possible of work that has been done with regard to implementing a first circuit design can be reused in implementing the second circuit design.

The task of implementing a complex electrical circuit design in physical circuitry can be a formidable one. For example, there are various types of general purpose, programmable circuit devices that can be used to implement a wide variety of circuit designs. Illustrative of these types of devices are programmable logic arrays ("PLAs"), field programmable gate arrays ("FPGAs"), and other similar programmable integrated circuit devices. (See Cliff et al. U.S. patent 5,260,611 and commonly assigned, co-pending U.S. patent application No. 08/442,795, filed May 17, 1995, for examples of PLA devices. These documents are hereby incorporated by reference herein.) Typically, in the use of these kinds of devices the user devises a logic circuit design for performing desired logic functions. This circuit-design phase may be computer

assisted (e.g., using a system such as the MAX+PLUS II Programmable Logic Development System, which is commercially available from Altera Corporation of San Jose, California, and which is described, for example, in the Altera publications "MAX+PLUS II Getting Started", Version 5.0, July 1994, and "MAX+PLUS II AHDL", Version 5.0, July 1994, both of which are hereby incorporated by reference herein.) Additional modules of this computer software are then frequently used to translate the user's circuit design into a set of components and inter-component connections that can be implemented in the physical circuitry of the device to be used to implement the design. For example, this process may involve breaking the user's circuit design into a plurality of components, each of which is implementable in a respective one of the physical circuit components of the intended device. (In this case a physical circuit component may be a logic gate, a logic module (e.g., circuitry capable of forming any logical combination of several inputs), or a logic array block (e.g., several interconnected logic modules).) In addition, all necessary connections between the circuit design components are identified for implementation in the interconnection resources of the intended device. Such a decomposition of the user's logic design may be embodied in a structured list of all required logic components and their interconnections. A list of this type is sometimes referred to as a "netlist," and that term is therefore employed herein with that meaning.

After a netlist has been generated, considerable further work must often be done -- again by other portions of the above-mentioned software -- to decide how the netlist circuitry will actually be

implemented in physical circuitry. This process may involve such steps as synthesis (i.e., determining how a unit of available physical circuitry will be programmed to implement each design component), partitioning (i.e., grouping the design components into clusters that fit into available physical circuit clusters without excessive interconnections being required between the clusters), and placing and routing (i.e., assigning each design circuit cluster to a particular respective physical circuit cluster so that required connections can be made between the clusters without any required connection being blocked by other required connections).

Because so much computational effort often goes into implementing a netlist, it would be desirable to be able to reuse as much of that work as possible when there is a change in the circuit design.

In view of the foregoing, it is an object of this invention to provide methods for implementing circuit designs in physical circuitry which reduce the need to re-perform such steps as circuit synthesis, circuit partitioning, and circuit placement and routing when the circuit design changes.

It is a more particular object of this invention to provide methods for comparing netlists of circuit designs before and after design changes in such a way that unchanged portions of the design are identified so that the work of implementing those portions of the circuit can be reused in implementing the changed design.

Summary of the Invention

These and other objects of the invention are accomplished in accordance with the principles of the

invention by comparing netlists of circuit designs before and after design changes. Components of the two circuit design netlists having the same user-assigned names are tentatively matched to one another. Each such tentative match is further tested by comparing certain basic features of the tentatively matched components. If the basic features are also the same, the tentative match is verified or made final, and the matched components are added to a set of components that are common to the two netlists.

After all components have been considered for matching as described above, the as yet unmatched components are further considered for possible matches by generating a signature for each as yet unmatched component based on certain basic features of that component. The same signature-assigning algorithm is used for both netlists. However, this signature-assigning algorithm avoids assigning the same signature to any two components within either netlist by (if necessary) additionally considering the basic features of components that are neighbors of the component being given a signature, and even (if further necessary) the basic features of components that are neighbors of those neighbor components. At a certain point, however, if uniqueness of signatures is not achieved by additionally considering neighboring components, the process of considering more remote neighbors stops and a random, non-reproducible signature is assigned to the component. After signatures have been assigned to all as yet unmatched components in both netlists, signatures from the two netlists that match are identified. The basic features of components with matching signatures are then compared, and if these features are also the same, these components are added

to the set of components that are common to the two netlists.

The components that have not been matched in either of the two preceding matching phases are then further considered. If such a component has a neighboring component that has a matching component in the other netlist, then the basic features of that component are compared to the basic features of components in the other netlist that neighbor the matching component in the other netlist. If a basic feature set match is found, the associated components are added to the set of components that are common to both netlists.

After the foregoing component-matching procedures have been completed, inter-component connections are considered. For each matched component, if a matched immediate neighbor of that component is not matched to one of the immediate neighbors of that component's match in the other netlist, then the connection between that component and that matched immediate neighbor is marked as changed.

The end result of the foregoing process is identification of (1) components that match, (2) components that do not match, and (3) changed connections. This identification is available for both the old and new netlists. Matched components in the old and new netlists represent unchanged parts of the netlists. Unmatched components in the old netlist represent deleted regions of that netlist. Unmatched components in the new netlist represent new regions in that netlist. Marked connections represent changed connections between unchanged components in the two netlists. This information can be used to identify those portions of post-netlist processing (performed

for the old netlist) that can be reused for the new netlist. Conversely, this information also identifies those portions of the post-netlist work that must be re-performed for the new netlist.

Further features of the invention, its nature and various advantages will be more apparent from the accompanying drawings and the following detailed description of the preferred embodiments.

Brief Description of the Drawings

FIGS. 1a-1g (collectively referred to as FIG. 1) are a flow chart of an illustrative embodiment of the methods of this invention.

FIG. 2a is a diagrammatic representation of information about an illustrative circuit design, which information can be used by the methods of this invention.

FIG. 2b is similar to FIG. 2a, but shows information for a modification of the FIG. 2a circuit design. This information is used with the FIG. 2a information by the methods of this invention.

FIGS. 3a and 3b are respectively similar to FIGS. 2a and 2b, but show another example of illustrative circuit design information which can be used by the methods of this invention.

Detailed Description of the Preferred Embodiments

For purposes of illustration, some of the data contained in two circuit design netlists is represented graphically in FIGS. 2a and 2b. Although netlists are typically structured data lists in a computer database, for present purposes it is easier to represent relevant aspects of that data graphically as in FIGS. 2a and 2b. FIG. 2a shows a netlist for an

"old" design, while FIG. 2b shows a netlist for a "new" design (i.e., a design which has substantial similarities to the old design, but also some differences from that old design). In each netlist each circle represents a "component" of the circuit design. For example, a component may be a logic gate or group of logic gates, a register, etc. Each straight line represents an inter-component connection in the circuit design. Each component has been given a name by the circuit designer or "user." For ease of reference herein, all names begin with the letter N (e.g., N1, N2, N3, etc.). Each component also has a set of basic features, identified herein by the shorthand references which begin with the letter F (e.g., F1, F4, F6, etc.). Any of a wide range of component parameters can be included among those that form the so-called basic feature set or characterization of each component. For example, the basic features may include the general type or function of the component, the number of connections it has to other components, etc. The basic features of the various components typically distinguish many components from one another, but they do not typically distinguish each component from all others. Thus it will be seen that in each of the netlists of FIGS. 2a and 2b some of the basic features (e.g., F1 and F6) appear in more than one component. Some of the components shown in FIGS. 2a and 2b may be inputs to and/or outputs from the depicted circuitry.

To briefly review the differences between the circuit designs represented by the netlists of FIGS. 2a and 2b, the components with names N1, N2, N3, N5, N8, N14, N22, and N25 appear to be unchanged from FIG. 2a to FIG. 2b. Each of these components has the same name

and the same basic features in both FIGS. The component with the name N4 appears to have changed because its basic features F4 in FIG. 2a become F9 in FIG. 2b. The component with the name N12 in FIG. 2a appears to have been eliminated from the FIG. 2b circuit design. The component name N19 in FIG. 2a is not reused in FIG. 2b, although that component may have been replaced by component N27 (note that components N19 and N27 have the same basic features). Component N30 appears to be new in FIG. 2b.

While the netlists shown in FIGS. 2a and 2b do not require use of all the features of the present invention, they do illustrate use of some of those features. Basically, the principal objective of the methods of this invention is to find portions of a new netlist (e.g., FIG. 2b) that are sufficiently similar to an old netlist (e.g., FIG. 2a) so that for those portions of the new circuit design the detailed work that has already been done to implement the old circuit design can be reused. New design implementation work will therefore only have to be done for those portions of the new circuit design that are not sufficiently similar to the old design. In present terms, circuit design components in two netlists that are basically the same as one another are said to "match."

Turning now to the illustrative embodiment of the methods of this invention which is shown in FIG. 1, the process begins with step 100 in which each component in the new netlist is considered in turn. (Step 100 could alternatively be performed by considering each component in the old netlist in turn, but the result will be exactly the same whichever netlist is treated as the reference list.) In step 100 the user-assigned name (e.g., N1, N2, N3, etc.) of each

component in the new netlist is searched for among the names of the as yet unmatched components in the old netlist in order to identify (in step 102) tentatively matching components in the old netlist. In this way components N1, N2, N3, N4, N5, N8, N14, N22, and N25 in FIG. 2b will be tentatively identified as respectively matching the similarly named components in FIG. 2a. Steps 102-106 are bypassed for any new netlist components whose user-assigned names are not found in the old netlist. Thus steps 102-106 will be skipped for components N27 and N30 in FIG. 2b.

In step 104 the basic features of the tentatively matched components are compared. If the basic features of the tentatively matched components are the same, step 106 is performed to finalize the matching of those components by adding them to a list of components that have been found to match between the old and mew netlists. Step 106 will be performed for components N1, N2, N3, N5, N8, N14, N22, and N25 in FIG. 2b. However, step 106 will be bypassed for component N4 in FIG. 2b because the basic features of that component are not the same in FIGS. 2a and 2b. The tentative matching of components N4 in FIGS. 2a and 2b will therefore be withdrawn when step 106 is bypassed for that component name.

Step 108 is a bookkeeping step which returns control to step 100 until appropriate ones of steps 100-106 have been performed for all components in the new netlist. Thereafter step 108 passes control to step 110.

Step 110 is another bookkeeping step which passes control to step 112 if any components in the new netlist remain unmatched after steps 100-108 have been performed for all new netlist components. On the other

hand, if no new netlist components remain unmatched, step 110 passes control to step 180. In the particular example illustrated by FIGS. 2a and 2b components N4, N27, and N30 remain unmatched when step 110 is performed, and so control passes from step 110 to step 112.

Step 112 begins a series of steps that attempt to match components based on their basic features (in the event that the name but not the basic features of a component have been changed). In step 112 a user-assigned variable MAXLEV is read to establish the maximum number of levels of neighboring components that will be considered in an effort to avoid assignment of the same signature to more than one component in either netlist. For example, if MAXLEV is 0, only the basic features of a component will be considered in assigning a signature to that component. If MAXLEV is 1, the basic features of a component and its immediate neighbor components may be considered in assigning a signature to the component (assuming that the component does not receive a unique signature based solely on its own basic features). (A component is an immediate neighbor of another component if there is a direct connection between the two components. For example, in FIG. 2a the immediate neighbors of component N1 are components N2, N3, N4, and N14.) If MAXLEV is 2, the basic features of a component, its immediate neighbors, and immediate neighbors of those immediate neighbors may be considered in assigning a signature to the component (assuming that the component does not receive a unique signature based on its own basic features and the basic features of its immediate neighbors). Still higher values of MAXLEV may be used if desired.

In step 114 an index variable ILEV is initialized to 0.

In step 120 the basic features of each as yet unmatched component in each netlist is used to assign a signature to that component. The signature assigning technique (an example of which is discussed somewhat later) is such that any components with the same basic features will be assigned the same signature, but components with different basic features will receive different signatures. Thus in step 120 components N4, N12, and N19 in FIG. 2a are assigned signatures based on their basic features F4, F7, and F10, respectively, and components N4, N27, and N30 in FIG. 2b are assigned signatures based on their basic features F9, F10, and F8, respectively.

After step 120 has been performed, step 122 is performed to determine whether any signature has been used more than once in either netlist. Ultimately it is desired for the signatures used for each netlist to be all unique. In the particular example shown in FIGS. 2a and 2b this will be the case for both netlists because the three components from each netlist considered in step 120 all have different basic features. Thus in this example control passes from step 122 to step 140. However, if for either netlist step 120 resulted in any signatures being duplicated, control would pass from step 122 to step 124 to begin a process for eliminating such duplicated signatures. For example, if component N30 in FIG. 2b had basic features F9 instead of F8, step 120 would result in duplicate signatures for components N4 and N30. Steps 124 et seq. would then be performed to eliminate this signature duplication.

In step 124 the index variable ILEV is incremented by 1.

In step 126 the current value of ILEV is compared to MAXLEV. If ILEV is greater than MAXLEV, control passes from step 126 to step 132, which will be discussed below. On the other hand, if ILEV is not yet greater than MAXLEV, control passes from step 126 to step 128.

In step 128 the signatures of any components with signatures that are duplicated in the same netlist are replaced by new signatures that are based on the basic features of that component and all components that are within ILEV neighbor levels of that component. Any signature assignment that is based in part on the basic features of neighboring components to a given number of neighbor levels is preferably biased toward being different from any signature assignment that is based on a lesser number of neighbor levels (to reduce the chances of creating new signature duplication).

After each performance of step 128, step 122 is performed again to repeat the test for signature uniqueness within each netlist. If any signature duplication is still present, steps 124 and 126 are repeated, and if MAXLEV is large enough, step 128 is performed again with a higher value for ILEV. Eventually, however, if ILEV exceeds MAXLEV without signature uniqueness being achieved within each netlist, then step 132 will be performed to just assign a random, non-reproducible signature to each component that has not otherwise received a unique signature within the netlist that includes that component. The signatures assigned in step 132 have a very high probability of being unique. For this reason, and because they are "forced," they are sometimes referred

to for convenience herein as "necessarily unique" signatures (although there may in fact be a tolerably small probability that they are not unique). These "necessarily unique" signatures effectively eliminate any component with such a signature from matching on the basis of signature in step 140.

An illustrative example of a signature assigning technique (usable in steps 120 and 128) includes the following steps:

1. output = some_fixed_value
2. output = random_num1 x (output ^ (feature x (neighbor_level + random_num2)))
3. go back to 1 for each feature of the component.

If ordering of the features does not matter, step 2 can be rewritten as follows:

2. output = (output ^ (feature x (neighbor_level + random_num2))

These steps are a hashing function that takes a numerical input and produces a numerical output. The input is a numerical representation of the basic features of a component (e.g., number of inputs/outputs, type of component, types of components at outputs, etc. The neighbor_level is an index value (similar to ILEV) up to the current value of ILEV. The symbols x, ^, and + indicate multiplication, EXCLUSIVE OR, and addition, respectively. All inputs and outputs are integer numerical values.

In step 140 an attempt is made to match each as yet unmatched component in the new netlist with each as yet unmatched component in the old n tlist on the basis of finding components in the two netlists with the same signature. If the same signature is found in

both netlists, step 142 is performed to tentatively match the associated components. Then in step 144 the basic features of the tentatively matched components are compared. If the basic features are the same, the tentative matching is finalized in step 146, and the matched components are added to the list of previously matched components. If the basic features are not the same, step 146 is bypassed, thereby discarding the tentative matching of these components. Step 148 is a bookkeeping step to return control to step 140 until all as yet unmatched components have been tested for matching on the basis of signature.

As in the case of step 100, the references in step 140 to the new and old netlists could be reversed with no change in the result.

In the particular example shown in FIGS. 2a and 2b performance of steps 140-148 will result in matching new netlist component N27 with old netlist component N19. Even though these components have different user-assigned names (i.e., N19 and N27), they have the same basic features and they are therefore matched in this phase of the method.

Step 150 is another bookkeeping step for determining whether the next phase of the method should be performed to continue to attempt to match components. If there are any still unmatched components in both netlists, control passes to step 160 to begin the next phase of the method. Otherwise control passes to step 180, thereby bypassing steps 160-170.

In step 160 each as yet unmatched component is considered to determine whether it has an immediately neighboring component that has been matched. If not, control passes to step 170 to begin

consideration of the next as yet unmatched component. On the other hand, if the as yet unmatched component has a matched immediate neighbor, control passes from step 160 to step 162.

In step 162 the match in the other netlist of the immediate neighbor identified in step 160 is identified. Then in step 164 the immediate neighbors of the component identified in step 162 are identified. Next, in step 166, the basic features of each component identified in step 164 are compared to the basic features of the component being considered from step 160. If the basic features of one of the components identified in step 164 are found to be the same as the basic features of the component being considered from step 160, then in step 168 those two components are matched and added to the list of previously matched components. If no commonality of basic features is found in step 166, step 168 is bypassed and no match is made.

Matching components in accordance with step 166 may be additionally conditioned on the order of the connections to the two components being considered for matching. For some types of circuitry the order of connections to components may not matter. For other types of circuitry the order of connections does matter. For circuitry of the latter kinds, the user may additionally require step 166 to ensure that the order of the connections to components being considered for matching is the same before identifying those components as matched.

Step 170 is a bookkeeping step to make sure that control returns to step 160 as long as it is sensible to continue looking for matching components

based on matching neighbors of as yet unmatched components.

FIGS. 2a and 2b do not require use of the portion of the invention which is the subject of steps 160-170. This portion of the invention tends to come into play for components whose signatures have to be based in part on basic features of neighboring components or whose signatures are forced to be random and non-reproducible (steps 124-132). While necessary in the signature matching phase of the invention, these forms of signature assignment can sometimes prevent matching that should take place. The search for matching components based on matched immediate neighbors (steps 160-170) can find such matches.

When step 180 is reached, the component-matching portion of the process is complete. In the particular example shown in FIGS. 2a and 2b, components N1, N2, N3, N5, N8, N14, N19, N22, and N25 in FIG. 2a will have been respectively matched with components N1, N2, N3, N5, N8, N14, N27, N22, and N25 in FIG. 2b. Components N4 and N12 in FIG. 2a and components N4 and N30 in FIG. 2b will be unmatched.

The final portion of the process (steps 180-190) relates to finding connections between matched components that have changed from the old netlist to the new netlist. In step 180 each matched component is considered in turn. Each matched immediate neighbor of the component being considered is identified. Then in step 182 the component in the other netlist which matches the component being considered is identified, as are the components in the other netlist which match the immediate neighbors of the component being considered. Still in step 182 it is det rmined whether each of the immediate neighbor matches is an immediate

neighbor of the match of the component being considered. If this test is not satisfied for any of the immediate neighbors identified in step 180, then in step 184 the connection between the component being considered and that immediate neighbor is designated as changed. Step 183 (instead of step 184) is performed for immediate neighbors for which the test of step 182 is satisfied. Step 183 designates the connection being considered as unchanged.

Step 182 is another step (like step 166) that may be additionally conditioned on the order of connections to components. Again, for some types of circuitry the order of connections to components may not matter, while for other types of circuitry the order of connections does matter. For the latter kinds of circuitry, the user may additionally require step 182 to ensure that the order of the connections to components being considered for changed connections be the same before identifying connections as unchanged.

Step 186 is a bookkeeping step which returns control to step 180 until all matched components in both netlists have been considered for possibly changed connections. Then the process ends at step 190.

FIGS. 3a and 3b illustrate the function performed by steps 180-186. FIG. 3a represents a fragment of an "old" netlist in which all of components 1-5 have been found to respectively match components 1-5 in a "new" netlist (a fragment of which is shown in FIG. 3b). In FIG. 3a the immediate neighbors of component 1 are components 2, 3, and 4. In FIG. 3b component 3 is no longer directly connected to component 1. Instead, component 5 is now an immediate neighbor of component 1. The performance of steps 180-186 for component 1 in the old netlist (FIG. 3a)

results in the connection between components 1 and 3 in that netlist being designated as changed. The performance of these steps for component 1 in the new netlist (FIG. 3b) results in the connection between components 1 and 5 in that netlist being designated as changed. This illustrates that even though none of the components may change between old and new netlists, connections between those components may change. Steps 180-186 are used to find those changed connections between unchanged components.

From the foregoing it will be seen that performance of the methods of this invention results in identification of matched components in both netlists, complementary identification of components in both netlists that do not match, and identification of changed connections in both netlists. Matched components represent unchanged portions of the old netlist. Unmatched components in the old netlist represent deleted portions of that netlist. Unmatched components in the new netlist represent new portions of that netlist. Connections identified as changed represent changed connections between unchanged components. This information is useful in reducing the amount of work that must be done to implement in physical circuitry the circuit design represented by the new netlist. For example, synthesis, partitioning, placing, and routing that has been performed for portions of the old netlist design that are not changed in the new netlist design can be reused rather than having to be done again "from scratch." Only for the new or changed portions of the new netlist design must all the implementation details be newly devised.

Computer software for implementing the methods of FIG. 1 can be included at the appropriate

point in computer software systems of the type mentioned above (e.g., the MAX+PLUS II system). Operationally, this software addition functions between (1) the circuit design and netlist creation modules, and (2) the modules for implementing netlists in physical circuitry. This software addition identifies the portion of previous work done by the latter modules that can be reused and therefore does not have to be redone to implement a new netlist.

It will be understood that the foregoing is only illustrative of the principles of the invention, and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention. For example, it has been pointed out in connection with several of the steps shown in FIG. 1 that references to the "old" and "new" netlists in those steps can be reversed with no change in the result. As another example of modifications within the scope of this invention, the term "netlist" is used herein as a convenient shorthand term for certain kinds of circuit design information. It will be understood that the methods of this invention are applicable to these kinds of information, even though that information may not be embodied in something called a netlist. For example, the circuit design information could be found in a graphical representation (like FIGS. 2a, 2b, 3a, or 3b) rather than in any form of list. As still another example of possible modifications, it may not be necessary to produce actual lists of matched and/or unmatched components. Instead, it may be sufficient to simply record a flag associated with each component indicative of whether the component is matched or unmatched. The flag can include the identification of the matching

component in the other netlist. It may not be necessary or desired in all cases to perform all phases that are included in the presently preferred embodiment shown in FIG. 1.

CLAIMS

1. A method for identifying portions of a new circuit design that are substantially similar to portions of an old circuit design, each of said designs including a plurality of components and a plurality of connections between various ones of said components, each of said components having a basic feature set, said method comprising the steps of; assigning signatures to components in each of said designs, each signature being indicative of the basic feature set of the component to which it is assigned; identifying any signature that is duplicated for a design; replacing the signature of any component, whose signature is identified as duplicated in said identifying step, with a new signature that is indicative of the basic feature set of said component whose signature is being replaced and the basic feature sets of components that are connected to said component whose signature is being replaced via paths made of said connections, none of said paths including more than a predetermined number of said connections in series; further identifying any signature that results from said replacing step that is duplicated for a design; effectively eliminating from further consideration on the basis of signatures any component whose signature is identified as duplicated in said further identifying step; and comparing the signature of each component having a signature in a first of said designs to the signatures of components having

signatures in a second of said designs in order to match components with the same signature.

2. The method defined in claim 1 further comprising the step of: further comparing the basic feature sets of each pair of components that have been matched in said comparing step in order to confirm the match of only those component pairs with the same basic feature set.

3. The method defined in claim 1, wherein in said predetermined number is two.

4. The method defined in claim 1, wherein said effectively eliminating step comprises the step of: further replacing the signature of any component, whose signature is identified as duplicated in said further identifying step, with a further new signature that is known to be unique for the signatures for that design.

5. The method defined in claim 1, wherein each said further new signature is a random, non-reproducible signature.

The Patent Office

23

Application No: GB 9811847.4
Claims searched: 1-5

Examiner: Mike Davis
Date of search: 8 September 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): G4H (HU), G4A (AUB)

Int Cl (Ed.6): G06F

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
	None	

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family.
A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.